APPLICATION FOR UNITED STATES LETTERS PATENT

APPARATUS AND METHOD FOR PERFORMING FAST FIBRE CHANNEL WRITE OPERATIONS OVER RELATIVELY HIGH LATENCY NETWORKS

Inventors:

Murali Basavaiah
916 Bonneville Way
Sunnyvale, CA 94087
Citizenship: India

Satish Ambati
147 N Milton Avenue
Campbell, CA 95008
Citizenship: India

Magesh Iyengar
3/A Krishi Vihar Colony
Indore, Madhya Pradesh
India. Pincode: 452 001
Citizenship: India

Thomas Edsall
13208 Peacock Court
Cupertino, CA 95014
Citizenship: USA

Dinesh G. Dutt
1176 Corral Avenue
Sunnyvale, CA 94086
Citizenship: India

Silvano Gai
3021 Mauna Loa Court
San Jose, CA 95132
Citizenship: Italy

Varagur V. Chandrasekaran
240 Estrella Road
Fremont, CA 94539
Citizenship: United States

Assignee:
Andiamo Systems, Inc
375 East Tasman Drive
San Jose, CA 95134
Entity: Large

Beyer Weaver & Thomas, LLP
P.O. Box 778
Berkeley, CA 94704-0778
Tel: (650) 961–8300

APPARATUS AND METHOD FOR PERFORMING FAST FIBRE CHANNEL WRITE OPERATIONS OVER RELATIVELY HIGH LATENCY NETWORKS

FIELD OF THE INVENTION

[0001] The present invention relates generally to network communications, and more particularly, to an apparatus and method for performing fast Fibre Channel write operations over relatively high latency networks.

BACKGROUND OF THE INVENTION

[0002] With the increasing popularity of Internet commerce and network centric computing, businesses and other organizations are becoming more and more reliant on information. To handle all of this data, storage area networks or SANs have become very popular. A SAN typically includes a number of storage devices, a plurality of Hosts, and a number of Switches arranged in a Switching Fabric that connects the storage devices and the Hosts.

[0003] Most SANs rely on the Fibre Channel protocol for communication within the Fabric. For a detailed explanation of the Fibre Channel protocol and Fibre Channel Switching Fabrics and Services, see the Fibre Channel Framing and Signaling Standard, Rev 1.90, International Committee for Information Technology Standards (INCITS), April 9, 2003, and the Fibre Channel Switch Fabric - 2, Rev. 5.4, INCITS, June 26, 2001, and the Fibre Channel Generic Services – 3, Rev. 7.01, INCITS, November 28, 2000, all incorporated by reference herein for all purposes.

[0004] The infrastructure of many networks often includes multiple types of link level transports. For example, the communication network of an international corporation may have local SANs in their New York, Silicon Valley and Tokyo offices respectively. However, since maintaining a SAN across long distances is expensive, the organization may rely on the Internet Protocol (IP) over another inter-SAN link such as Gigabit Ethernet, SONET, ATM, wave division multiplexing, etc., to connect the SANs.

[0005] Within a typical SAN with Fibre Channel Inter-Switch Link (ISLs), the access time between a Host and a storage device (i.e., a target) is typically very fast. The speed of a Fibre Channel link is such that the performance or access time across multiple switches in close to the ideal, i.e., the Host and the target device are attached to the same switch. In other words, even if multiple Switches need to be spanned to complete the access, the speed of the individual Switches is so fast that the latency time is typically very small. In a write operation for example, packets of data can be transferred across the switches of the SAN without delay as the latency between the ISLs is very small.

[0006] In situations with a high latency inter-SAN link, however, the access time of a write operation between a Host in one SAN and a storage device in a remote SAN will suffer or deteriorate. The latency may result from the speed of the link, the distance between the Host and target, congestion on the inter-SAN link, etc. For example, when IP is used to connect two Fibre Channel SANs, the latency across the IP portion of the network is typically slow relative to an access within the SANs.

[0007] With a SCSI write command, the Host will issue a write (Wr) command defining a certain amount of data to be written. The command travels across the network, from switch to switch, until it reaches the target. In reply, the target responds with a Xfer ready command which defines the amount of data which the target may accept. When the Host receives the Xfer ready command, it transfers the data to be written in units up to the maximum transfer unit (MTU) of the network. In most Fibre Channel SANS, the MTU is approximately 2K bytes per transfer. Thus if the amount of data to be written is 8K bytes, then a total of four transfers are required. When in this case all four data transfers are received, the target generates a status success command. If for some reason the Host does not receive the status command after a predetermined period of time, it is assumed that a problem with the write operation occurred. The Host may subsequently issue another write command.

[0008] The time required to complete a SCSI write operation can be significant over a high latency inter-SAN network. A significant amount of time may lapse between the time the initial Wr command is issued and the Xfer ready is received by the Host due to the slow performance of the high latency inter-SAN network. During this time, the Host is idle and must wait until before issuing the data transfer commands to transfer the data to the Host. The target is also idle until it receives the data from the initiating Host. In other words, the

initiating Host is idle until it receives the Xfer ready and the target is idle after issuing the Xfer ready until it receives the data.

[0009] An apparatus and method improving the performance of a SCSI write over a relatively high latency network is therefore needed.

SUMMARY OF THE INVENTION

[0010] To achieve the foregoing and other objectives and in accordance with the purpose of the present invention, an apparatus and method to improve the performance of a SCSI write over a high latency network is provided. The apparatus includes a first Switch close to the initiator in a first SAN and a second Switch close to the target in a second SAN. In various embodiments, the two Switches are border switches connecting their respective SANs to a relatively high latency network between the two SANs. In addition, the initiator can be either directly connected or indirectly connected to the first Switch in the first SAN. The target can also be either directly or indirectly connected to the second Switch in the second SAN. During operation, the method includes the first Switch sending Transfer Ready (Xfr_rdy) frame(s) based on buffer availability to the initiating Host in response to a SCSI Write command from the Host directed to the target. The first and second Switches then coordinate with one another by sending Transfer Ready commands to each other independent of the target's knowledge. The second switch buffers the data received from the Host until the target indicates it is ready to receive the data. Since the Switches send frames to the initiating Host independent of the target, the Switches manipulate the OX_ID and RX_ID fields in the Fibre Channel header of the various commands associated with the SCSI Write. The OX_ID and RX_ID fields are manipulated so as to trap the commands and so that the Switches can keep track of the various commands associated with the SCSI write.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the present invention may best be understood by reference to the following description of the presently preferred embodiments together with the accompanying drawings.

Figure 1 is a diagram of a high latency network connecting a Host in a first SAN and a storage device in a second SAN.

Figures 2A-2D are SCSI Command, Data, Response and Transfer Ready frames respectively.

Figure 3 is a diagram of a Fibre Channel header.

Figure 4 is a temporal diagram illustrating a SCSI fast write operation over a high latency network according to one embodiment of the present invention.

Figure 5 is a temporal diagram illustrating a SCSI fast write operation over a high latency network according to another embodiment of the present invention.

Figure 6 is a block diagram of a switch according to the present invention.

Like reference numbers refer to like elements in the figures.

DETAILED DESCRIPTION OF THE INVENTION

[0011] Referring to Figure 1, a diagram of a high latency inter-SAN network 10 connecting a Host H1 in a first SAN 12 and a target storage device T1 in a second SAN 14 is shown. The Host H1 is coupled to the high latency network 10 through a first switch SW1 in SAN 12. The target storage device T1 is coupled to the network 10 through a second switch SW2. The switches SW1 and SW2 are considered "border" switches since they are positioned at the interface of the network 10 and the SANs 12 and 14 respectively. According to various embodiments, the Host H1 and target T1 may be either directly connected to switches SW1 and SW2 or connected indirectly through any number of intermediate switches respectively. The network 10 may use the Internet Protocol (IP) for example over an inter-SAN link such as Gigabit Ethernet, SONET, ATM, wave division multiplexing, etc. to connect the SANs 12 and 14. Again, the network 10 may have a high latency relative to the SANs 12 and 14 for a variety of reasons, such as the speed of the link, congestion on the link, or distance.

[0012] The present invention is related to a SCSI write operation that improves or reduces the time required to perform a write operation between the initiating Host H1 and target storage device such T1 over a high latency network such as the inter-SAN network 10. The Intelligent Ports (I-ports) of the two switches SW1 and SW2 act as an intermediary between the Host H1 and the storage device T1. The transfer size of a data transfer during a write operation is negotiated before any write operations are performed. Initially, the Host H1 defines (i.e., specifies the amount of data it wishes to write) the transfer size for a write command. The switch SW1 indicates the amount of data it is ready to receive based on (i) the data size specified in the Write command and (ii) the amount of buffer space it has. The I-port on SW1 responds with a Transfer Ready (Xfer) which indicates the maximum size of a data transfer. The I-port on the switch SW2 similarly receives the Xfer ready which defines the maximum size of the data transfer. In the aforementioned embodiment, the ports involved are Intelligent Ports (I-Ports) to which the initiator and target are attached. In such a case, the I-port is typically a FC port also sometimes referred to as an Fx_Port. In an alternative embodiment, the target and the initiating Host are not directly connected to the Switches in question. In such a case, the I-port can be either an IP-port or an I-port.

[0013] In general, the fast write operation is performed after the initial negotiation by the following sequence: (i) when the Host H1 generates a SCSI write command defining the target T1, the I-port of Switch SW1 traps the command; (ii) the switch SW1 forwards the

command to the target; (iii) the switch SW1 also issues a Transfer Ready command to the Host H1 on behalf of or as a proxy for the target T1; (iv) the Host H1 sends data of the amount indicated by the Transfer Ready amount to the target T1 in response to the received Transfer Ready command. The data may sequenced or broken up into frames based on the maximum transfer unit (MTU) of the network; (v) the I-port of the switch SW1 receives the data frames and forwards it to the target T1; (vi) the previous two steps are repeated until all the data is transferred to the target; and (vii) after all the data is transferred, the switch SW1 waits for either a success or error status command from the target T1. Upon receipt, the switch SW1 forwards the status command back to the Host H1. If the target returns an error command, no attempt is made by the I-port to correct the error. In should be noted that in an alternative embodiment, the above sequence can be performed by switching the order of steps (ii) and (iii) as defined above.

[0014] The I-port of the second switch SW2 operates essentially the same as switch SW1 except that it buffers the received data frames until receiving a Transfer Ready command from the target T1. Specifically, the I-port of switch SW2: (i) forwards the SCSI write command received from switch SW1 to the target; (ii) issues a Transfer Ready command to the switch SW1 as a proxy for the target T1; (iii) buffers the data frames received from the switch SW1; (iv) transfers the data frames to the target T1 when a Transfer Ready command is received from the target T1; and (v) after all the data is transferred, the switch SW2 waits for either a success or error status command from the target T1. Upon receipt, the switch SW2 forwards the status command back to switch SW1. If the target returns an error command, no attempt is made by the I-port of switch SW2 to correct the error.

[0015] To identify an FC device, Fibre Channel Identifiers (FCIDs) are used. A transaction between an FC host and a target is referred to as an exchange. In a typical Fibre Channel network, there are many Hosts and targets. Each Host may initiate many read and/or write operations. For the hosts and targets within a network to keep track of the various transactions between each other, two fields are available in the Fibre Channel header for all SCSI Command, Data, Response, and Transfer Ready frames. The first field is called the Originator Exchange Identifier or OX_ID. The second field is called the Receiver Exchange Identifier or RX_ID. The Host relies on the OX_ID to maintain its local state and the target relies on the RX_ID to maintain its local state. In both cases, the OX_ID and RX_ID are typically 16 bits wide.

[0016] The OX_ID and RX_ID are typically used by the initiating host and target of a transaction respectively to keep track of the ongoing transactions between the two entities. In general, the switches in a SAN do not keep track of such transactions. With the present invention, however, the switches SW1 and SW2 are acting as intermediaries between the initiating Host and the target T1. The switches SW1 and SW2 therefore also use the OX_ID and RX_ID values to track exchanges between the Host H1 and the target T1.

[0017] Referring to Figures 2A-2D, SCSI Command, Data, Response and Transfer Ready frames are shown respectively. As illustrated in Figure 2A, the SCSI command frame includes a FC header field 20, a SCSI header field 22, and a FC-CRC field 24. As illustrated in Figure 2B, the SCSI Data frame includes a FC header field 20 and a data field 26. As illustrated in Figure 2C, the SCSI Response frame includes a FC header field 20 and a response frame 28. As illustrated in Figure 2C, the SCSI Transfer Ready frame includes a FC header field 20 and a transfer ready (Xfr-rdy) field 30.

[0018] Referring to Figure 3, a diagram of a Fibre Channel header field 20 is shown. The frame includes an OX_ID field 32 and an RX_ID field 34 along with a number of other fields (which are labeled in the figure but not described herein for the sake of brevity). As previously noted, the OX_ID field 32 and the RX_ID field 34 are each 16 bits wide and are used for identifying the originating Host and target device. Since each of the above-identified SCSI frames includes a header field 20 with an OX_ID field 32 and an RX_ID field 34, the switches in a Fibre Channel network can track of a given SCSI exchange between the identified originating Host and target device.

[0019] Referring to Figure 4, a temporal diagram is shown illustrating a SCSI write operation between the Host H1 in SAN 12 and target T1 over a inter-SAN network 10 according to the present invention. In the diagram, the direction of the arrows shows the flow of frames during the write operation. The vertical column, from top to bottom, indicates the passage of time. When a SCSI write operation is performed between the Host H1 and the target T1, the following sequence of events occur:

a. Host H1 initiates the fast write operation by issuing a SCSI write command (Wr: OX_ID = 1 RX_ID = 0Oxffff, Size = 10MB). The command defines the originating exchange identifier as 1 (OX_ID = 1). The receiving exchange identifier RX_ID, however, is "uninitialized" and is set to a default value of "Oxffff". The write command also specifies the amount of data to be written,

which in this example, is 10 megabytes (MB).

b. Upon receipt, the switch SW1 initializes the receiving exchange identifier RX_ID. In this example, the RX_ID is initialized to 10. The switch SW1 then determines if it has sufficient storage space to buffer the data. Assuming that it does, the switch SW1 sends a Transfer Ready command (Xrdy: OX_ID = 1, RX_ID = 10, Size = 10 MG) to the Host H1. All subsequent commands or frames between the Host and switch SW1, and vice versa, associated with this SCSI write operation define the OX_ID = 1 and the RX_ID =10. If the switch SW1 does not have sufficient buffer space, then a SCSI busy status is returned to the host H1, mimicking the behavior of a target when the target does not have resources for a new exchange.

c. The initiating switch SW1 uses the OX_ID to keep track of the transaction. Consequently, the switch SW1 changes the OX_ID provided by the initiating Host H1. In this example, the switch SW1 changes the OX_ID value to 10. The switch SW1 then forwards the write command to the target T1 with the RX_ID value remaining uninitialized (Wr: OX_ID = 10, RX_ID = Oxffff, Size = 10MB). All communication between the first switch SW1 and the target involving this write operation thereafter includes an OX_ID = 10 and RX_ID = Oxffff. The initiating switch SW1 uses the OX_ID value as a handle or pointer into a session table 36 maintained at switch SW1. The table includes an entry that includes information regarding the session that is accessed by the RX_ID handle.

d. When the second switch SW2 receives the write command, it initializes an exchange identifier entry in the sessions table 38 and it immediately forwards the command to the target T1 provided the switch SW2 has sufficient buffer space. If it does not have sufficient space, then a SCSI busy status is sent back to the initiating host H1.

e. If the target T1 is ready to receive the data, it sends a Transfer Ready command back to the switch SW2. According to one embodiment, the target designates an RX_ID value for the write transaction. In this case, the target designates an RX_ID value of 50. The Transfer Ready command received by the switch SW2 therefore appears as (Xrdy: OX_ID = 10, RX_ID = 50, size =

10MB). All subsequent communications between the switch SW2 and the target T1 involving this transaction include OX_ID value of 10 and an RX_ID value of 50. The switch SW2 also maintains a sessions ID table 38. Upon receipt of the Transfer Ready command, the switch SW2 inserts a RX_ID = 50 value into the table. The switch SW2 uses the modified OX_ID = 10 value as a handle or pointer into a sessions ID table 38. The target switch SW2 uses the OX_ID value as a handle or pointer for this session between in session table 38. The table includes an entry that includes the information regarding the session such as the target RX_ID.

f. If the second switch SW2 receives the data frames (Wdata: OX_ID = 10, RX_ID = Oxffff) from the first switch SW1 before the Transfer Ready command from the target T1, then the second switch SW2 buffers the data. When the Transfer Ready command is received, the data frame(s) are then forwarded to the target T1. On the other hand, if the data frames arrive after the Transfer Ready command, the data frames are immediately forwarded to the target T1.

g. When all the data has been transferred, the target T1 generates a Status command (Status: OX_ID=10, RX_ID=50). The second switch SW2 modifies the RX_ID = Oxffff and forwards the status command to the switch SW1. The switch SW1 in turn changes the RX_ID = 10 and sends the status command to the Host H1 to complete the fast write operation. It should be noted that in the event the target T1 provides a transfer size less than the requested size, the I-port on the switch SW2 waits for successive Transfer Ready commands until the requested size is met1.

[0020] It also should be noted that the Switches SW1 and SW2 "trap" Extended Link Service or ELS frames (state management frames) that contain the original OX_ID and RX_ID in the payload since the switches change the original values of OX_ID and RX_ID. ELS frames are used by the initiator H1 and target T1 to query and manage state transactions, such as ABTS and REC ELS frames.

[0021] Referring to Figure 5, an alternative embodiment of the present invention is shown. With this embodiment, the RX_ID, command frame Wr and the Transfer Ready frame Xry are used by the switches SW1 and SW2 to communicate with one another regarding buffer

availability and allocation for a transaction. In Figure 5 for example, the switch SW1 uses the RX_ID = 10 value in the Wr command (Wr: OXID = 10, RXID = 10 MB, size = 10 MB) to (i) specify the amount of buffer space needed for the write transaction; and (ii) use the command frame to request the needed buffer space. The switches also use the Transfer Ready frame to grant buffer space for the transaction. In this example, the switch SW2 generates a first Transfer Ready command with 5MB encoded in the RX_ID value (Xrdy: OX_ID = 1, RX 5 MB). The issued Transfer Ready command indicates to the switch SW1 that 5MB have been allocated for the write transaction. The switch SW1 consequently sends up to 5MB to switch SW2. When a second 5MB of buffer space becomes available, a second Transfer Ready command is issued (Xrdy: OX_ID = 1, RX_ID = 10, Size = 10 MB). Note, the RX_ID value for the second command is set to 10MB, indicating that the accumulative or total allocated buffer space for the transaction is 10MBs. The second Transfer Ready indicates that the remaining 5MB of buffer space is now available.

[0022] In an alternative embodiment, it is possible for switch SW2 to grant more buffer space than requested by SW1. Based on the previous example, SW2 could grant 15 MB instead of 10 MB. The remaining unutilized buffers are used for subsequent Write commands from the Host H1. For example, consider a second Write command for say 1 MB from the Host H1. With this embodiment, SW1 would send a Xfr_Rdy for 1 MB to the Host H1 and send the command to the target via SW2 as stated in paragraph 0021. When the Host H1 sends data, SW1, instead of waiting for Xrdy_Rdy to come from SW2 before sending data, now immediately starts transferring the data to SW2. It can do this because SW2 had previously granted additional buffers to SW1 via the last Xrdy_Rdy command. The basic idea is that the data can be transferred from SW1 to SW2 for subsequent (after the first) write commands without waiting for a specific Xrdy_Rdy from SW2 pertaining to the subsequent write.

[0023] In various embodiments of the invention, a number of alternatives may take place in situations where the switch SW1 has no available buffer space. In one embodiment, the Host H1 receives a busy status signal and the Host must re-try the write transaction; In a second embodiment, the command is placed in a pending command list. Eventually, the switch SW1 responds to the write but only after the processing the preceding transactions on the list. In yet another embodiment, the switch SW1 can simply forward the Write command to the target.

[0024] In yet another embodiment, the switches SW1 and SW2 are configured to set the Burst Length and Relative Offset fields in the Transfer Ready frame both to zero (0). This enables the other switches to differentiate if the Transfer Ready command was generated by the target switch or the target itself. The initiating switch and Host realizes that the target switch issued the Transfer Ready when both fields are set to zero (0) since the target itself would never set both to zero for a given transaction. If only one or neither of the fields are set to zero, then the initiating switch SW1 and Host realizes the Transfer Ready was generated by the target.

[0025] In data networks, data frames are lost on occasion. In various embodiments of the present invention, an one of a number of different buffer credit recovery schemes may be used.

[0026] Referring to Figure 6, a block diagram of a switch SW according to the present invention is shown. The switch 40 includes a data plane 42 and a control plane 44. In the data plane 42, the switch includes switching logic 46 connected between two sets of ports (including the I-ports) 48a and 48b. The switching logic 46 is configured to route or internally switch traffic received on one port 48a to another port 48b and vice versa. The control plane 44 includes a processor 50 for implementing all the switching Fibre Channel functionality and protocols such as those specified in the aforementioned INCITS documents, incorporated by reference herein, the Fibre Channel adapted versions of OSPFv3, IS-IS and/or BGP4+ routing protocols, or any other protocols useful for either intra-Switch or inter-switch communication. In various embodiments of the invention, the processor 50 may be implemented in a state machine, a micro-controller, hardware, firmware, programmable logic, or a combination thereof. As previously noted, the I-ports of the switch 40 negotiate with the initiating host the amount of data that can be transferred by a Write command (Wr) without waiting for a Transfer Ready command command from the target. This negotiation can takes place, for example, when the initiating Host issues a PLOGI or a PRLI to the target storage device. After the negotiation, the I-ports of the initiating and target switches SW1 and SW2 set up hardware filters to trap the any SCSI Write Commands between the specified Virtual SANs (VSANs) and initiating Host FC_ID and target device FC_ID. The trap is based on a tuple defined by VSAN, Host FC_ID, target FC_ID, OX_ID and RX_ID of the frame. Whenever a command defining the specified tuple is received, the command is

trapped by the switch. The term "trap" has used herein means the frame is not forwarded its destination, but rather is provided to the processor 50 of the switch for further processing.

[0027] When a Write command is received at the initiating switch SW1 that specifies a tuple to be trapped, the switch SW1 forwards it to the processor 50. In reply, the processor 50 is responsible for forwarding the original frame on to the original destination and generating a Transfer Ready command to the initiating Host H1. The Transfer Ready command defines a data size determined by the existing buffer space at the switch SW1. The processor also defines the locally generated RX_ID which is used for all subsequent communication between the switch SW1 and the initiating Host H1. When the data frame is received from the Host H1 at the I-port of the switch SW1, the frame is trapped. The processor 50 in turn instructs the switch SW1 to transmit the data frames up to the negotiated size without waiting to receive a Transfer Ready command. Any remaining claims are buffered. Similarly, at the I-port of the switch SW2, any data frames associated with this exchange are trapped and buffered. When a Transfer Ready is received from the target T1, the switch SW2 transfers the buffered data.

[0028] Transfer Ready frames involving this exchange received by either switch SW1 and SW2 are also trapped and forwarded to the processor 50. The target switch SW2 uses the Transfer Ready frame to start the transfer of data to the target. The initiating switch SW1 on the other hand, uses the TransferReady command to transmit more data frames toward the target. In either case, the I-ports of both switches SW1 and SW2 modify the RX_ID's.

[0029] According to one embodiment, the Fibre Channel cyclical redundancy check or CRC included in the Fibre Channel header 20 is recomputed to protect rewrite operations. The CRC protects FC payload and FC header from corruption while traversing various parts of a Fiber Channel SAN. With the present invention, the RX_ID and OX_ID fields are modified, the FC headers need to be protected and the CRC recomputed to protect the rewrites from any corruption.

[0030] Although only a few embodiments of the present invention have been described in detail, it should be understood that the present invention may be embodied in many other specific forms without departing from the spirit or scope of the invention. Therefore, the present examples are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein but may be modified within the scope of the appended claims.